Traffic Alert and Collision Avoidance System (TCAS) Program Office

Airborne Collision Avoidance System X (ACAS X) Overview

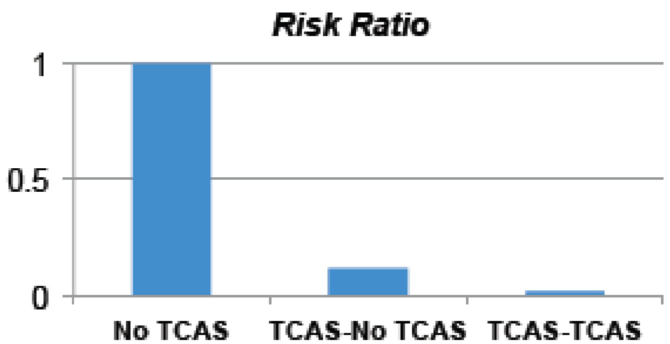
By: Mike Castle

Date: June 10-11, 2014

Mission Need (TCAS II Performance Review)

Safety

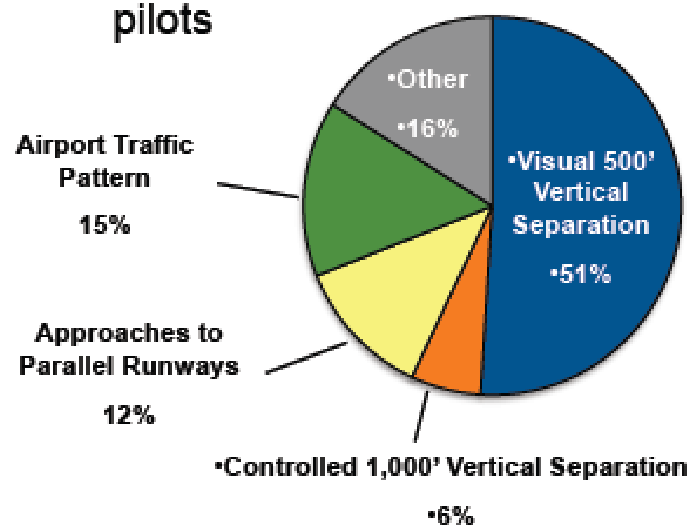
- No U.S. commercial air carrier collisions since mandate
- Numerous TCAS saves
  - *"…TCAS saved our lives." - Pilots*
- Mid-air collision risk reduced by 90%



Operational Suitability

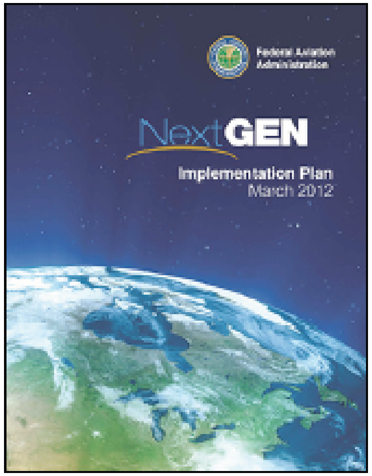
> 80% of alerts occur during intentional, safe operations

- Most cause minimal disturbance to pilots

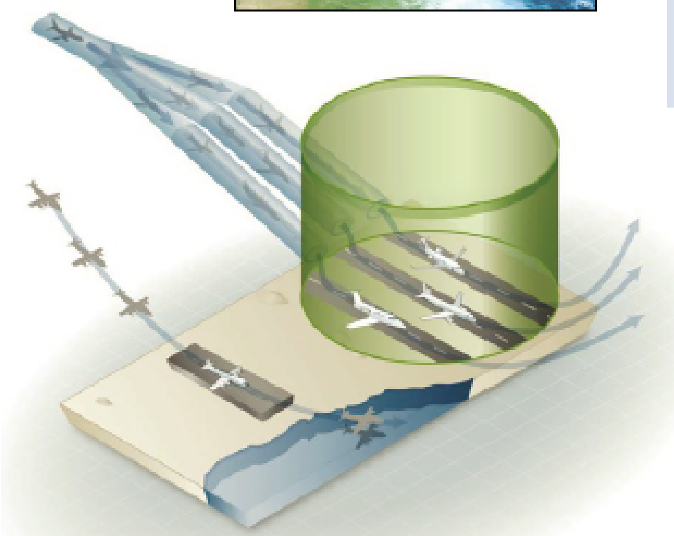


Performance monitoring assessment shows that TCAS works as intended but alerts during many normal, safe operations

Challenges for TCAS II in the Future

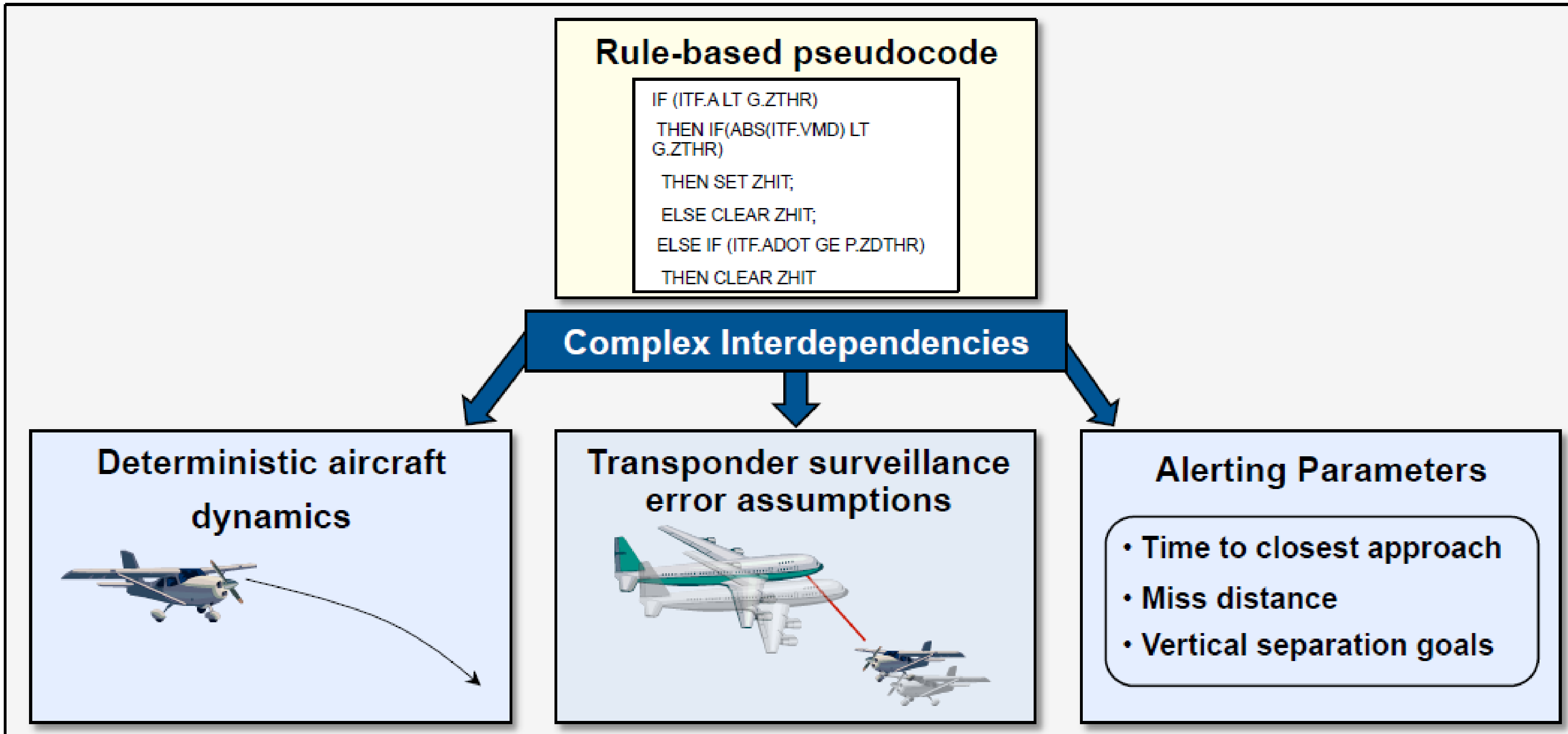


Future Airspace	TCAS Challenges
Additional surveillance information available	Tightly tuned to transponder-based surveillance, difficult to incorporate ADS-B
Reduced procedural separation	Alerting thresholds difficult to adjust without compromising safety
CAS for other user classes	Alerting logic not easily adaptable beyond current TCAS users



- **TCAS will not easily support new demands for flexibility and efficiency**
- **Difficult to accommodate new user classes such as general aviation and unmanned systems**

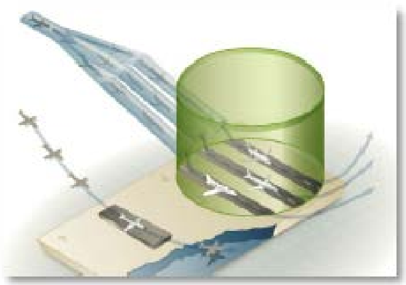
TCAS Upgrade Challenges



- Pseudocode is compilation of deterministic rules and heuristics
- Alerting criteria is tightly coupled to transponder surveillance performance
- Modifying alerting criteria or rules to address specific performance issues is difficult due to interdependencies

ACAS X Introduction

Airborne Collision Avoidance System



- ACAS X – An interoperable expansion of a family of aircraft collision avoidance systems developed for use in NextGen airspace
- Provides the same general role as TCAS II:
  - Surveillance of nearby aircraft
  - TA/RA Generation
  - Coordination with other aircraft collision avoidance systems

- Supports New Capabilities:
  - Leverages Additional Surveillance Sources
  - Intended for multiple types of host aircraft
  - Tunable for Reduced Separation Operations

ACAS X Program

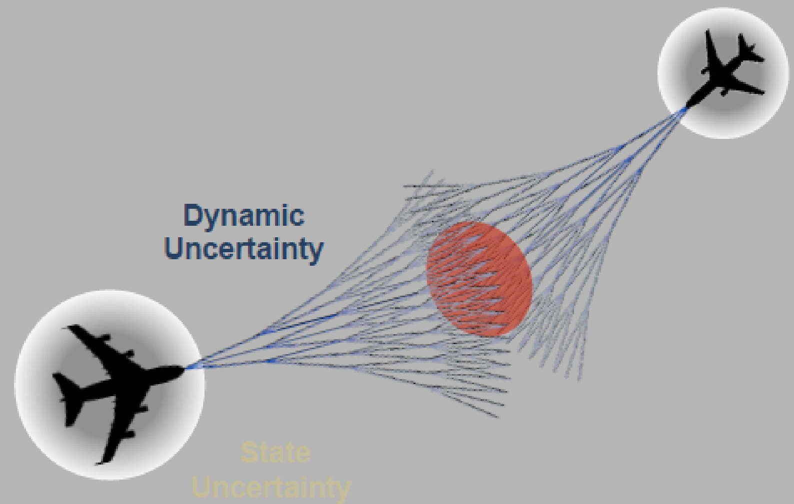
Airborne Collision Avoidance System (ACAS X)

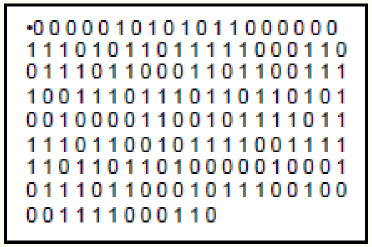
- **FAA initiated formal research in 2009**
  - Decision theoretic safety logic
  - Flexible surveillance tracker

- **Benefits**
  - Enables reduced separation
  - Fewer unnecessary alerts
  - Extends to new user classes
  - Easier to adapt to changing airspace

ACAS X supports NextGen airborne collision avoidance requirements

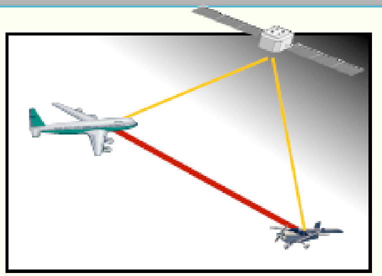
Next Generation System





Optimized Logic Table

(no pseudocode)



Plug-and-Play Surveillance

(sensor agnostic)

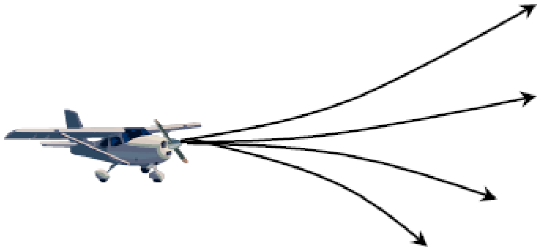
ACAS X Threat Logic

State Uncertainty

Imperfect sensor information leads to uncertainty in position and velocity of aircraft

Probabilistic sensor model

Dynamic Uncertainty



Variability in pilot behavior makes it difficult to predict future trajectories of aircraft

Probabilistic dynamic model

Multiple Objectives

System must carefully balance both safety and operational considerations

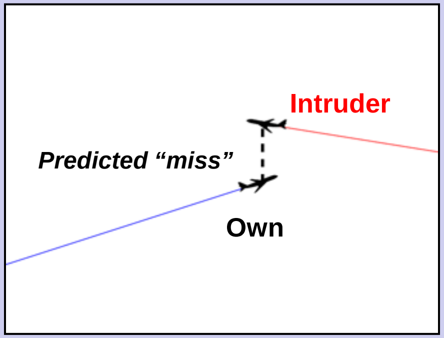
Multi-objective utility model

Optimal logic produced from probability and utility models

ACAS X Alerting is Different Than TCAS

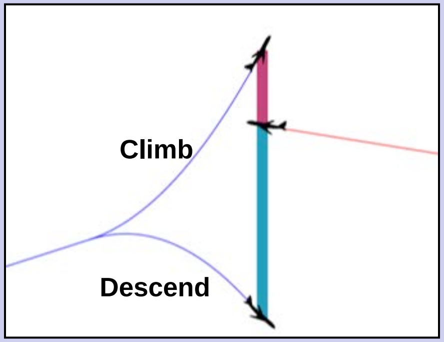
Legacy TCAS

Step 1: Is there a hazard?



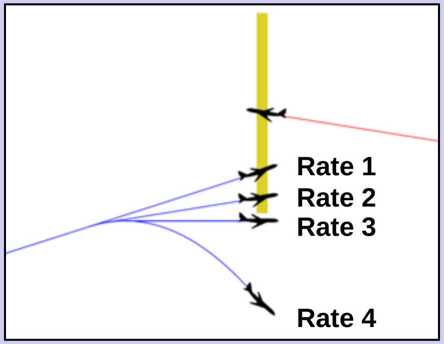
Alert if time to closest approach and projected miss distance are below thresholds

Step 2: Climb or descend?



Choose sense that maximizes miss distance

Step 3: What altitude rate?



Choose the lowest vertical rate predicted to achieve required separation

ACAS X

Step 1: State Distribution

Estimate ~10M states based on beliefs about own and intruder dynamics

Step 2: Look-Up Table

```
•0 0 0 0 0 1 0 1 0 1 0 1 1 0 0 0 0 0 0
1 1 1 0 1 0 1 1 0 1 1 1 1 1 0 0 0 1 1
0 0 1 1 1 0 1 1 0 0 0 1 1 0 1 1 0 0 1
1 1 1 0 0 1 1 1 0 1 1 1 0 1 1 0 1 1 0
1 0 1 0 0 1 0 0 0 0 1 1 0 0 1 0 1 1 1
1 0 1 1 1 1 1 0 1 1 0 0 1 0 1 1 1 1 0
0 1 1 1 1 1 1 0 1 1 0 1 1 0 1 0 0 0 0
0 1 0 0 0 1 0 1 1 1 0 1 1 0 0 0 1 0 1
1 1 0 0 1 0 0 0 0 1 1 1 1 0 0 0 1 1 0
```

For each state estimation, look up expected cost related to available actions

Step 3: Choose action with lowest cost

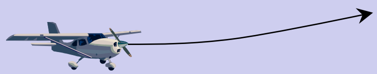
No Alert	0.8
Level-off	0.1
Descend	0.9
Climb	0.2

Choose the action resulting in the lowest cost

How ACAS X and TCAS Alerts are Modified

Legacy TCAS

Assumed behavior



Change assumptions of own and intruder aircraft behavior

Thresholds

- **Tau =** ~~40 sec~~ *35 sec*
- **Zthr =** ~~600 ft~~ *400 ft*
- **ALIM =** ~~300 ft~~ 250 ft

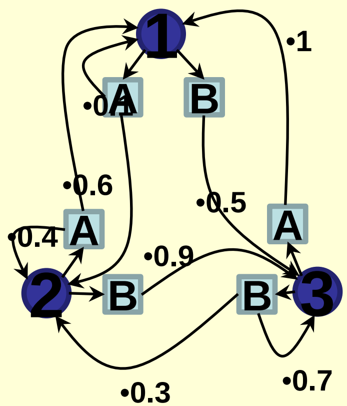
Establish new alert criteria

Rules

Change existing pseudo code

ACAS X

Dynamics



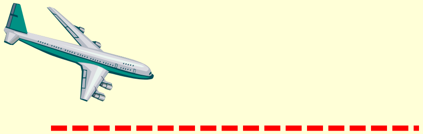
Modify weights of belief states and state transitions

Offline costs

- NMAC (-1)
- Alert (-0.01)
- Reversal (-0.01)
- Strengthen (-0.009)
- Clear of conflict (0.0001)

Change the reward values for alerting parameters

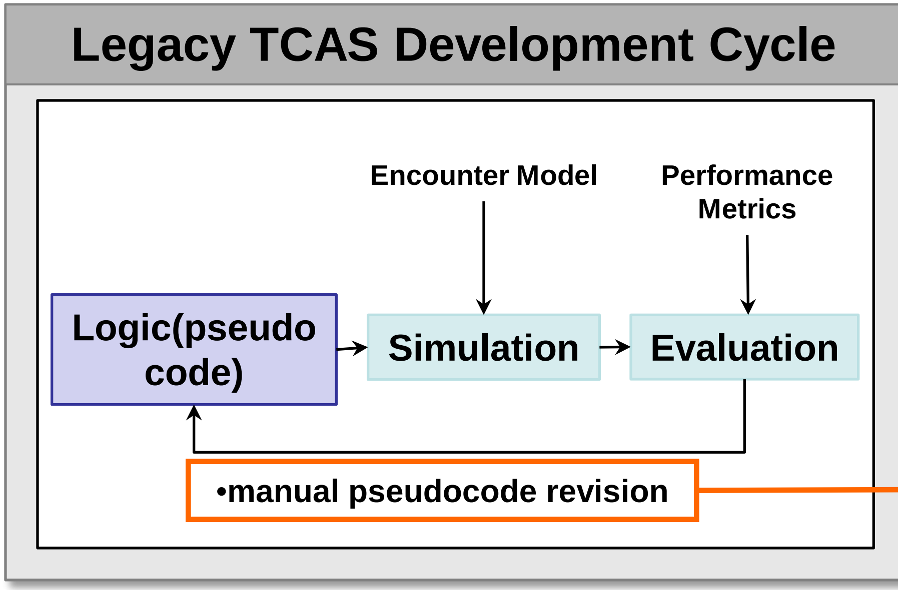
Online costs



Alert inhibit altitude

Used actual parameters live on aircraft

TCAS Logic Development

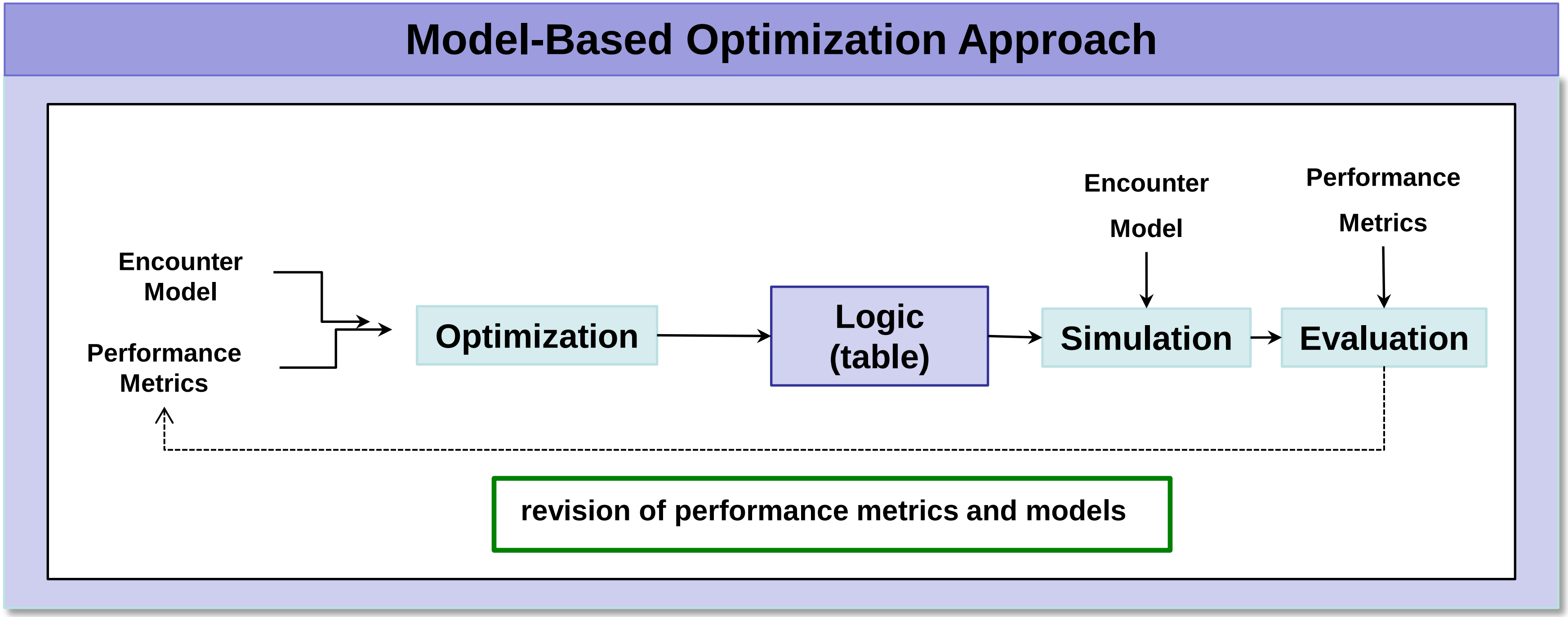



```
PROCESS Reversal_modeling;

.    NOMINAL_SEP = 0;
.    Z = G.ZOWN;
.    ZD = G.ZDOWN;
.    DELAY = 0;

.    IF (G.OWN_FOLLOW EQ FALSE)
.    .    THEN   CALL MODEL_SEP
.    .    .    .    .    IN (DELAY, ZD, Z, ZD, P.VACCEL, OWNTENT(7), ITF.ZINT, ITF.ZDINT, ITF entry)
.    .    .    .    .    OUT (NOMINAL_SEP);
.    .    .    .    IF (OWNTENT(7) EQ $TRUE)
.    .    .    .    .    THEN   NEW_SENSE = $FALSE;
.    .    .    .    .    ELSE   NEW_SENSE = $TRUE;
.    .    .    .    IF (NOMINAL_SEP GT 1.2 * P.CROSSTHR)
.    .    .    .    .    THEN   CLEAR ITF.REVERSE;
.    .    ELSE
.    .    <Begin own is assumed to follow its RA>
.    .    .    IF (OWNTENT(5,6) EQ '00')
.    .    .    .    THEN   DELAY = MAX(P.TV1 - (G.TCUR - G.TPOSRA), 0);
.    .    .    .    .    .    IF (OWNTENT(7) EQ $FALSE)
.    .    .    .    .    .    .    THEN   ZDGOAL = MAX(MIN(G.ZDOWN, P.MAXDRATE), P.CLMRT);
.    .    .    .    .    .    .    ELSE   ZDGOAL = MIN(MAX(G.ZDOWN, P.MINDRATE), P.DESRT);
.    .    .    .    .    .    CALL PROJECT_VERTICAL_GIVEN_ZDGOAL
.    .    .    .    .    .    .    IN ((G.TCUR - G.TPOSRA), G.ZTV, G.ZDTV, ZDGOAL, P.TV1, P.VACCEL)
.    .    .    .    .    .    .    OUT (ZPROJ, ZDPROJ);
.    .    .    .    .    .    IF   (((OWNTENT(7) EQ $FALSE AND ZPROJ GT G.ZOWN AND
.    .    .    .    .    .    .    (G.ZDOWN GE G.ZDTV - P.MODEL_ZD)) OR
.    .    .    .    .    .    .    (OWNTENT(7) EQ $TRUE AND ZPROJ LT G.ZOWN AND
.    .    .    .    .    .    .    (G.ZDOWN LE G.ZDTV + P.MODEL_ZD))) AND
.    .    .    .    .    .    .    G.TCUR – G.TPOSRA LT P.MODEL_T)
.    .    .    .    .    .    .    THEN   Z = ZPROJ;
.    .    .    .    .    .    .    .    .    ZD = ZDPROJ;
.    .    .    .    .    .    CALL MODEL_SEP
.    .    .    .    .    .    .    IN (DELAY, ZDGOAL, Z, ZD, P.VACCEL, OWNTENT(7),
.    .    .    .    .    .    .    .    ITF.ZINT, ITF.ZDINT, ITF entry)
.    .    .    .    .    .    .    OUT (NOMINAL_SEP);
.    .    .    IF (OWNTENT(7) EQ $TRUE)
.    .    .    .    THEN   NEW_SENSE = $FALSE;
.    .    .    .    ELSE   NEW_SENSE = $TRUE;
.    .    .    DELAY = MAX(P.TV1 - (G.TCUR - G.TLASTNEWRA), P.QUIKREAC);

.    .    .    IF (NEW_SENSE EQ $FALSE)
.    .    .    .    THEN   ZDGOAL = MAX(P.CLMRT, MIN(G.ZDOWN, P.MAXDRATE));
.    .    .    .    ELSE   IF (G.NODESCENT EQ $TRUE)
.    .    .    .    .    .    .    THEN   ZDGOAL = 0;
.    .    .    .    .    .    .    ELSE   ZDGOAL = MIN(P.DESRT, MAX(G.ZDOWN,        P.MINDRATE));
.    .    .    IF (G.REV_CONSDRD EQ FALSE)
.    .    .    .    THEN   IF ((ITF.INT_CROSS EQ $TRUE) OR (ITF.ZDINT EQ 0 AND
.    .    .    .    .    .    .    ITF.RZ GT 0) OR (ITF.ZDINT * G.ZDMODEL LT 0))
.    .    .    .    .    .    .    THEN   MZDINT = ITF.ZDOUTR;
.    .    .    .    .    .    .    ELSE   MZDINT = ITF.ZDINR;
.    .    .    .    ELSE   MZDINT = ITF.ZDINT;

.    .    .    CALL MODEL_SEP
.    .    .    .    IN (DELAY, ZDGOAL, Z, ZD, P.RACCEL, NEW_SENSE, ITF.ZINT, MZDINT, ITF entry)
.    .    .    .    OUT (ZMP);

.    .    .    IF (ZMP LE 0 OR NOMINAL_SEP GE G.ALIM)
.    .    .    .    THEN   CLEAR ITF.REVERSE;
.    .    <End own is assumed to follow its RA>
END Reversal_modeling;
```

- Human effort focused on pseudocode

- Time-consuming process
- Many parameters require tuning
- Unlikely to be optimal

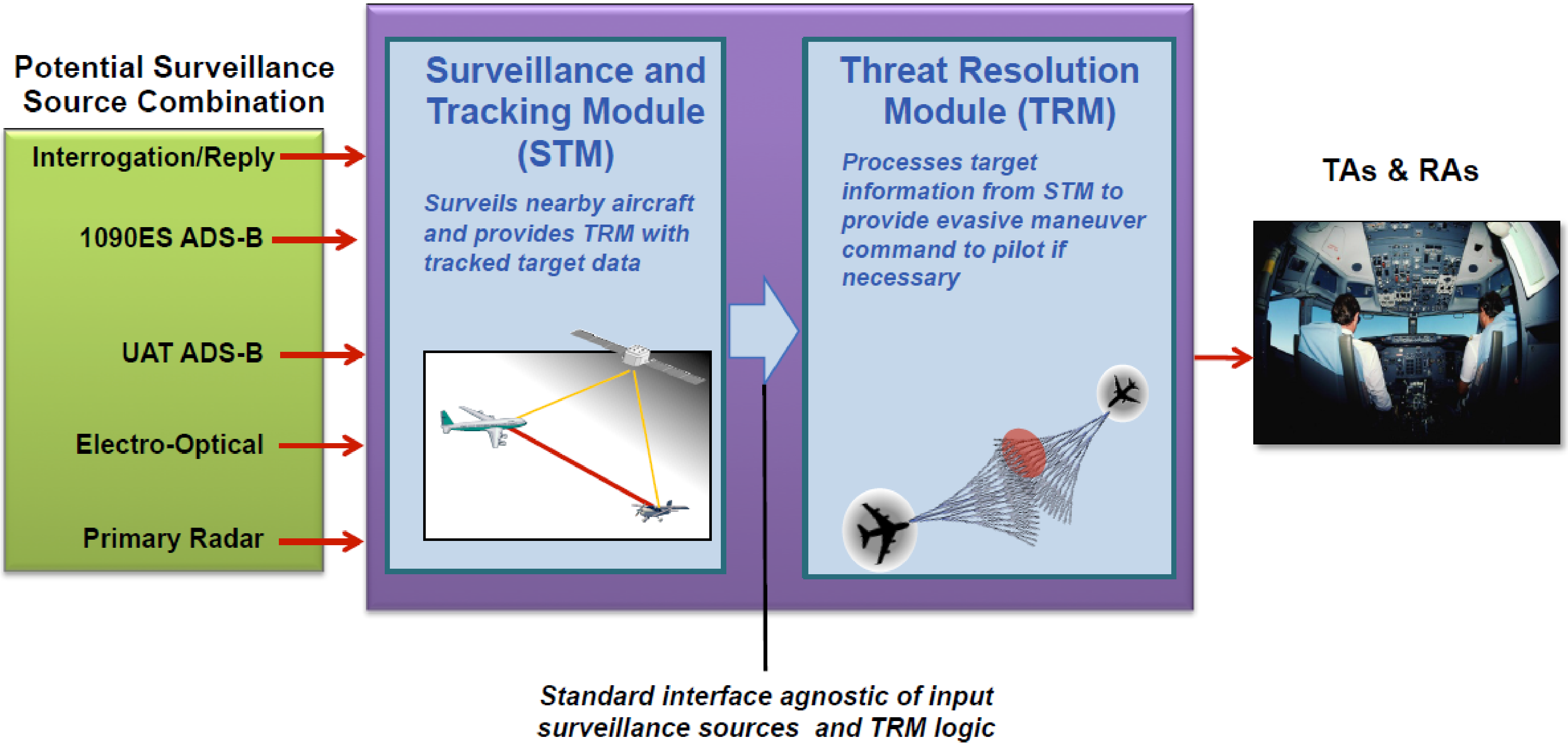
ACAS X Logic Development



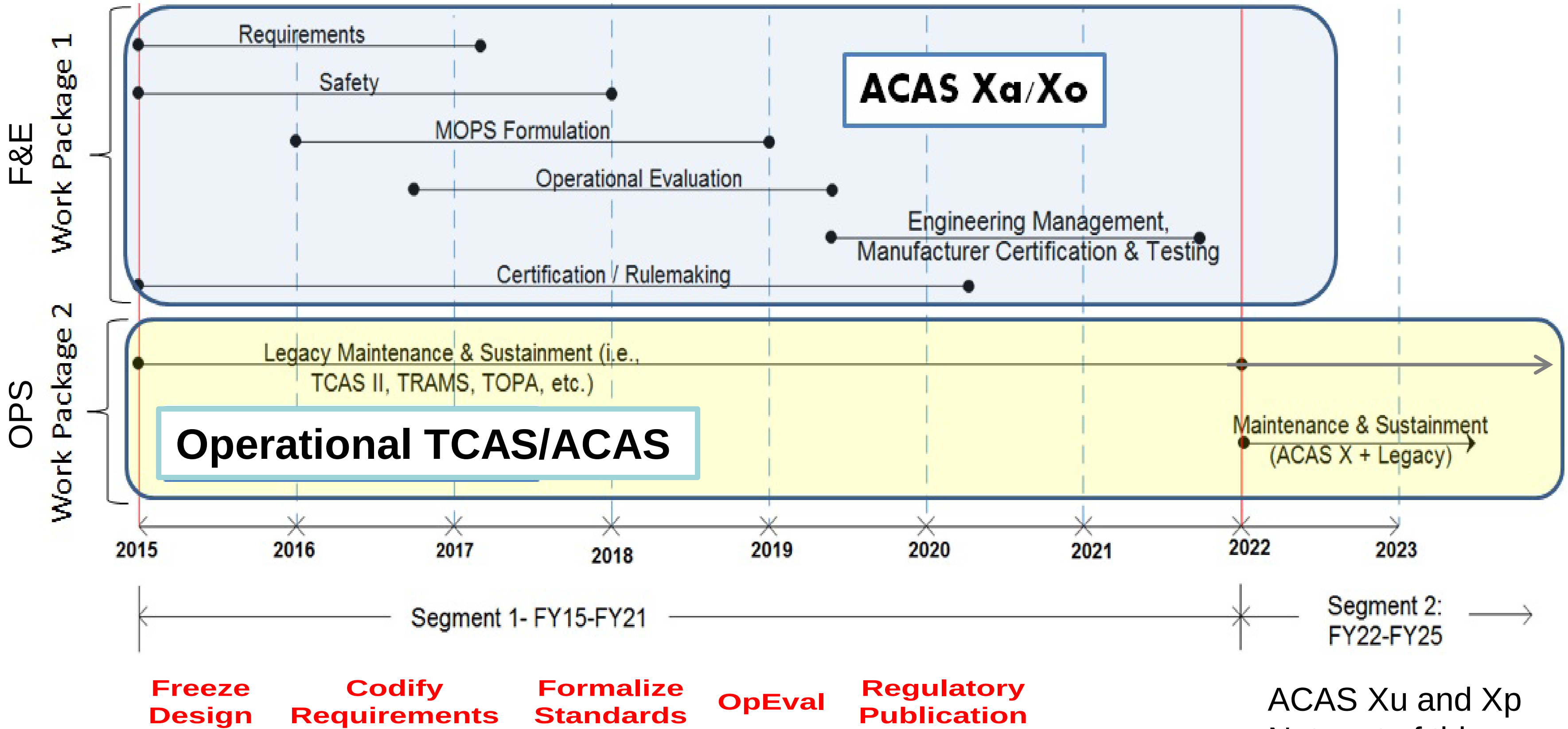
- Human effort focuses on defining performance metrics

- Computers generate lookup table
- Optimal, robust logic

ACAS X Architecture

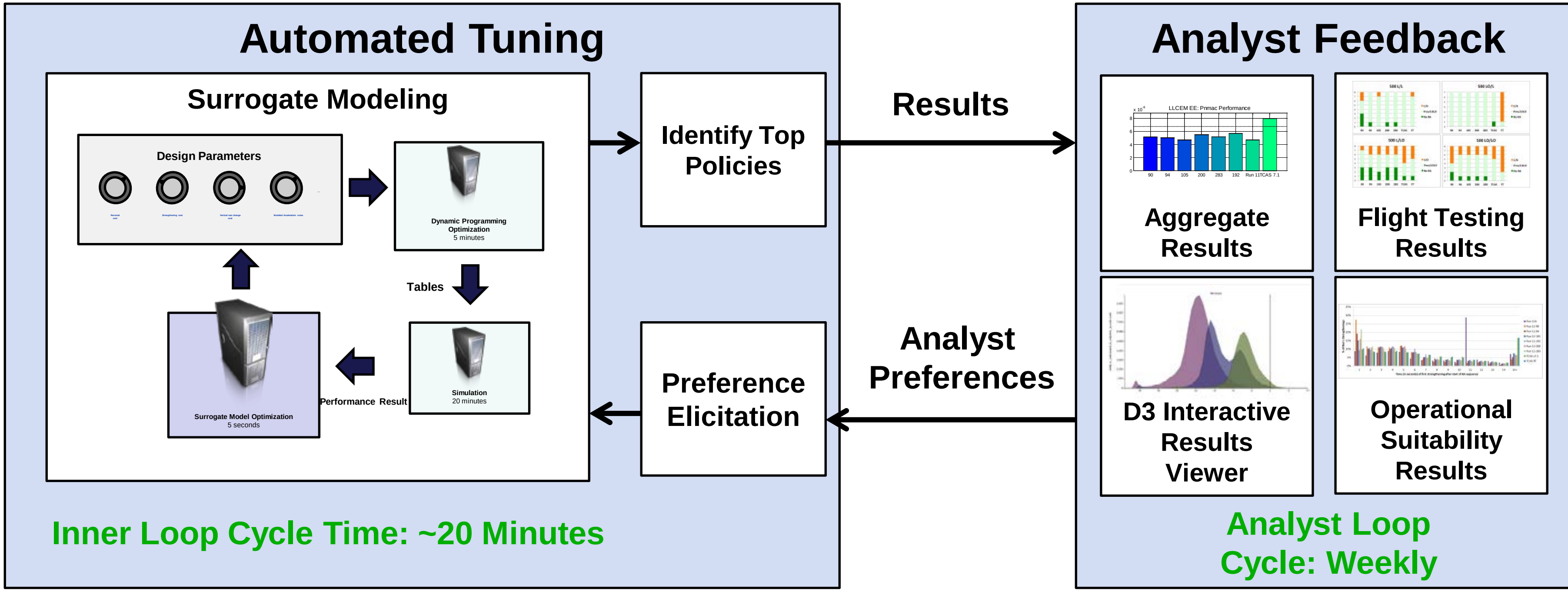


Program Segments and Work Packages



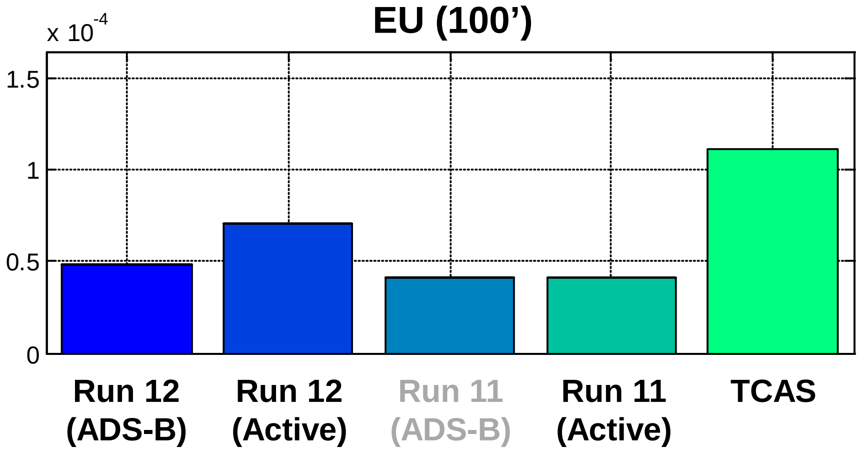
ACAS Xu and Xp Not part of this investment

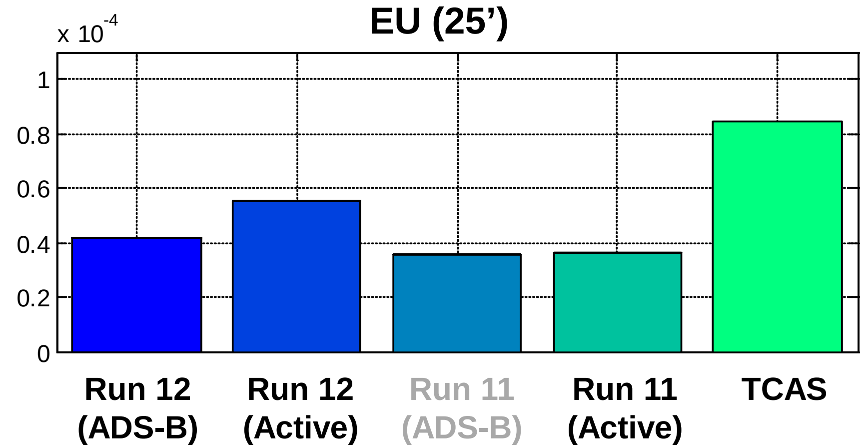
ACAS-X Tuning Process

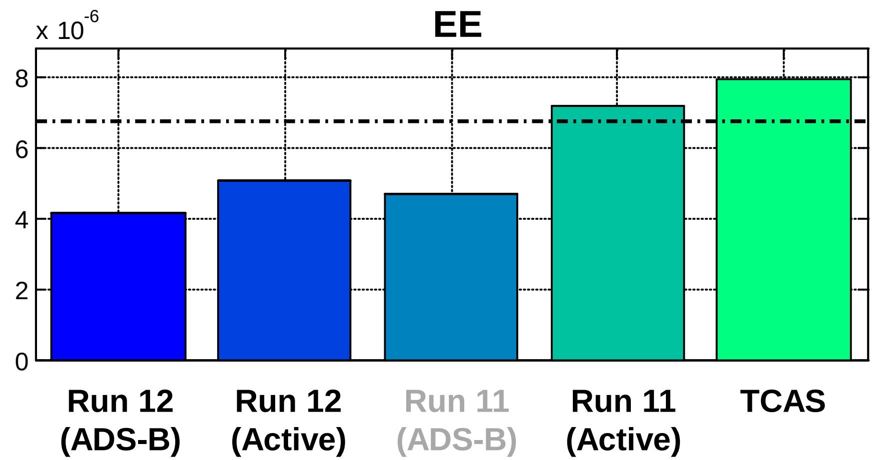


- **ACAS-X tuning accomplished via supervised optimization**
  - Analysts specify initial objective function and automated tuning performs search producing candidate logics
  - Automated search is periodically interrupted and analyst preferences incorporated via an updated objective function

Safety Results



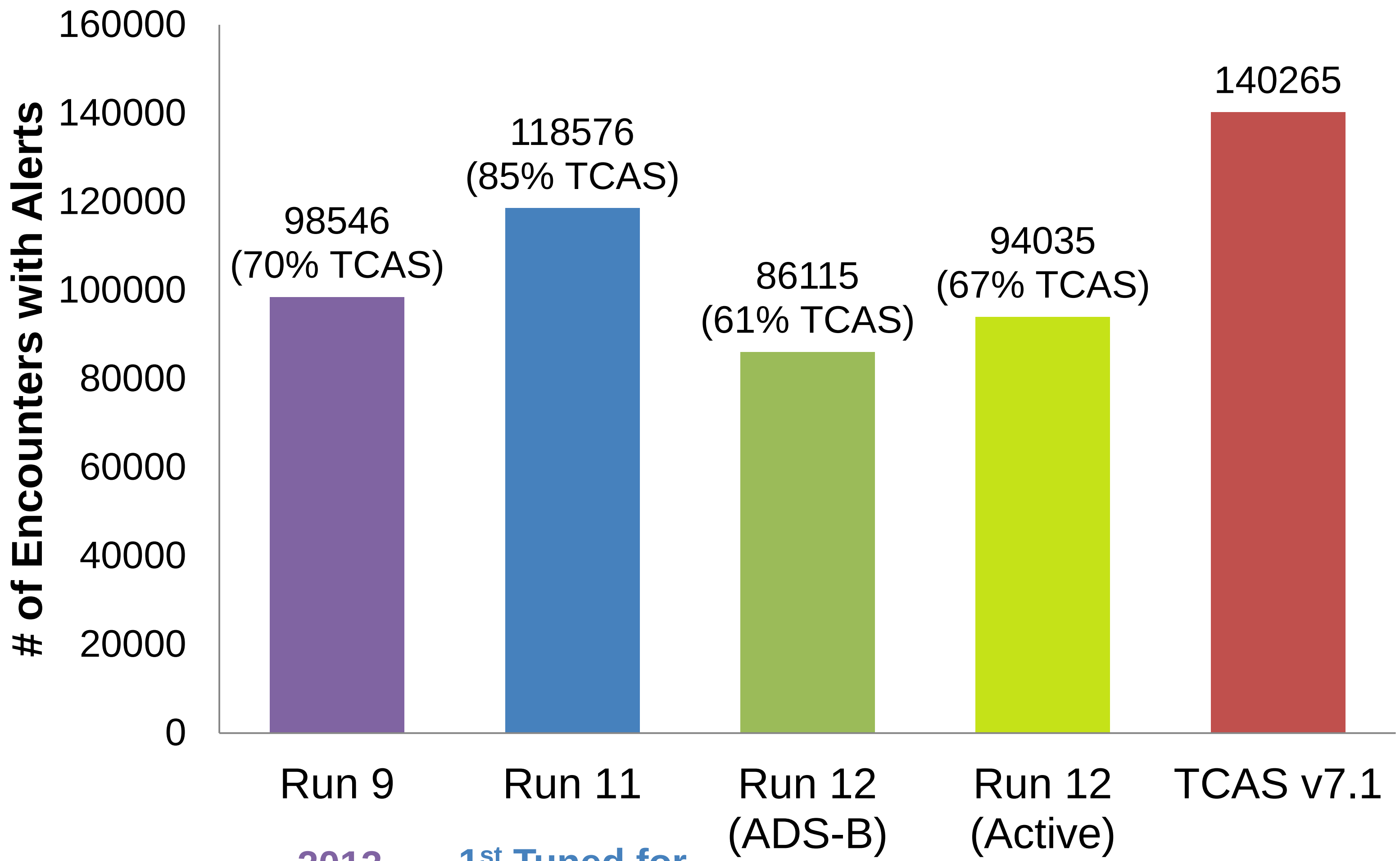




EE: Run 12 performance improves upon Run 11, exceeds TCAS
EU: Run 12 performance exceeds TCAS

Overall Alert Rate

12



2013 Flight Test

1st Tuned for Altimetry Bias

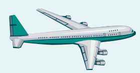
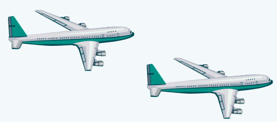
1st Tuned for ADS-B

Includes CSPO, all military, formation

BACKUP

X_A – **A**ctive Surveillance
X_O – **O**peration Specific
X_U – **U**nmanned Aircraft System
X_P – **P**assive Surveillance

ACAS X Variants

		User Group	Surveillance Technology	Advisories
Useful Segment 1	 ACAS X_A	Current TCAS II users (large aircraft)	Active radar supplemented with passive	Same as current TCAS II
Useful Segment 1	 ACAS X_O	Users of specific operations (e.g., CSPO, Formation Flights, ASAS Operations)	Active radar supplemented with passive	Procedure-specific alerts for selected aircraft, global alerting against all others
NextGen Concept Maturation	 ACAS X_U	Unmanned aircraft (controlled airspace)	Potentially radar, EO/IR, etc.	"Coordinated" vertical advisories
NextGen Concept Maturation	 *ACAS X_P +	General aviation, etc.	Passive only	Reduced advisory set